CHAPTER – 4
SOFTWARE RISK EVALUATION BASED ON DAG FORMATION

4.1 INTRODUCTION

Requirements engineering is the step in the software system development project that concentrates about the requirements of stakeholders. It involves the process of eliciting, structuring, specifying, elaborating, analyzing, and managing the problems of software solution that are required to be solved. Requirement engineering is the measured one of the most critical processes in the software development life cycle. A software project won’t be successful, if the goals of developing the software system, the requirements of stakeholders are not fulfilled. Consequently, ensuring the accuracy and completion of software project requirements is the goal of requirement engineering.

Each software development project has unique difficulties, both in terms of customer expectation and project features. A project has certain uncertainties that include the risk in requirement development, imprecise assessment of budget, high system complexity and continuous change of project scope to the latest market demand. During the life cycle of the software product, these factors raise the chance of potential risks. The software requirement engineering domain is more horizontal to risk, while comparing with other engineering domains. These risks possibly include:

i. Economic loss
ii. Lead to delays
iii. Customer dissatisfaction
iv. Reduce the ability to successful project completion
v. Market waste

Software risk management is used as an effective tool to achieve these risks. It is difficult to measure the concrete risk management advantage, even if the project succeeds. This work deals with the software development risk management modeling framework. It supports assessment and management of risks connected to the development environment,
operational constraints, project stakeholders and project execution before the elicitation of the system requirement.

Figure 4.1 Flow of the Proposed Method
Software risk management technique is a proactive risk management method that effectually contributes for a successful software project completion. The flow of the proposed method is shown in figure 4.1.

The research work is focused on coordinating and handling the large number of stakeholders’ requests and arranges the successive requirements into meaningful structures. The stakeholders are clustered based on the similarity among them. The relevant stakeholder’s clusters are used for software model generation. Directed Acyclic Graphs (DAG) technique is used to optimize the software requirement. Risk methodology provides a conceptual framework for risk management using a stakeholder oriented approach. It attempts to manage risks by capturing the intentions of stakeholders in the risk management process. It helps project managers to deliver the product with accurate and timely dissemination of project information for different stakeholders, thereby enabling to make critical decisions for the overall success of the project.

4.2 STAKEHOLDERS REQUIREMENTS

The requirement engineering process sometimes delivers lower quality and highly expensive software products. Customers are highly dissatisfied with such systems. Some of the delicate issues in requirements engineering include:

- Handling the relationships between requirements
- Managing changes to established requirements
- Managing dependencies between the requirements document and extra documents created during the systems and software engineering process.

These specifications should clearly and unambiguously reflect the preferred end result of both the customer and the developers in the final software product. According to this development, the system specifications are analyzed, planned, stored, controlled and maintained through the complete software development lifecycle. Therefore, it may be easily accessible at any given time by the stakeholders.

4.3 STAKEHOLDER REQUIREMENT ANALYSIS

Requirements analysis is a significant step in software development. Therefore, the requirement engineering process throughout the software development lifecycle has been essential in developing a successful solution.
A successful solution is attained when the customers' needs are established on time and within the allocated budget. An important feature of handling the requirements is to develop good communication among all the stakeholders involved throughout the software lifecycle. The beneficiaries of requirement engineering are the project stakeholders, developers, project manager and end users. Furthermore, the tester confirms the requirement status and changing the requirement, functionality and costs. Change in the stakeholder’s expectations sometimes leads to the removal of the requirements from the requirement list. In addition, it would directly affect the success of the project. Hence, recognizing potential variations is a very significant step in requirement engineering.

Stakeholder analysis needs to recognize and prioritize the requirements, and do not scale to large projects with many stakeholders and requirements. Some of the steps in stakeholder analysis include:

- Planning the process
- Defining and selecting a policy
- Recognizing key stakeholders
- Adapting the tools
- Recording and collecting the information
- Evaluating the stakeholder table
- Using the information

### 4.4 CLUSTERING OF STAKEHOLDERS

Several similarity measures are used while applying the clustering techniques. Yet, only few similarity measures are efficient to administer with the data analysis. In the proposed work, the similarities among the stakeholders are defined based on the Jaccard similarity measure. The resulting clusters of stakeholders’ requirements provide accurate and well-organized requirements to model the software. This section is described in chapter - 3.

### 4.5 DIRECTED ACYCLIC GRAPHS (DAG)

A graph-based modeling technique has been developed for the analysis of software comprising concurrency. DAG represents event-precedence networks, where activities may occur successively, probabilistically. When a set of activities occur concurrently, an
activity can be executed when all of its preceding activities are completed. This includes the special cases that one or all of the activities must be complete.

![Directed Acyclic Graph Representation](image)

Figure 4.2 Directed Acyclic Graph Representation

A significant subset of graphs is DAG. DAG method is used to describe the engineering procedure and systems. Generally, it consists of nodes and directional relationship among nodes. Then, a system $G$ can be described as follows,

$$G = (N, D)$$

Where, $N$ is defined as a set of nodes, $D$ is a set of relationship among nodes, $N_1 \rightarrow N_2$ means $N_2$ is executed only after $N_1$ is completed. The relationship among one node does not point itself directly and indirectly. In a directed acyclic graph, nodes stands for the activity to be executed, the relationship views for the relation between the activities and it can be an event data with certain specification. Activity execution reacts to a relationship and run some specific software component, in which software component is called as activity component. By using activity component, there exists a relationship between node and data and the software model is updated.

4.5.1 DAG MODEL PROCESS

The software requirement engineering can identify each software model to recognize the active component. In general, it can use the input and output data to identify and to build the activity component set, $A = (a_1, a_2, a_3, a_4, \ldots)$. 

80
Moreover, this process defines the relationship among the related activity component. Then, this technique can categorize the input and output data for all pairs of activity component and to build the relationship data set \( R = (r_1, r_2, r_3, r_4, \ldots) \). Directed acyclic graph creates the software model, which is based on the activity component and data set. For each software model, it needs to enhance this approach, until it completely meets the system requirement engineering, design and development requirements.

### 4.5.2 COMPONENT BASED SOFTWARE MODEL

According to the DAG model process; this work can build a dynamic model on DAG with activity component and data. In this model, it uses an activity component rather than action, as action is separate and non-controllable. The activity component is a combination unit that can be manageable, upgradeable and changeable. If some action’s working procedure is changed, it’s only needed to upgrade the corresponding activity component. Subsequently, it can develop the software model dynamically.

![Figure 4.3 Process of Directed Acyclic Graph](image-url)
4.6 CORRELATION MATRIX PREDICTION

There is a selection of common structures, which may be appropriate to use the working correlation matrix. If the number of observations per cluster is small in a composed and complete design, then an unstructured matrix is suggested. Additionally, for datasets with improper measurements, it may be reasonable to consider a model where the correlation is a function of the time between observations.

Moreover, comparison of approximations and standard error from different correlation structure may specify sensitivity to misspecification of the variance structure. The example of fixed working correlation matrix is shown below.

\[
R = \begin{pmatrix}
1.0 & 0.9 & 0.8 & 0.7 \\
0.9 & 1.0 & 0.9 & 0.8 \\
0.8 & 0.9 & 1.0 & 0.9 \\
0.7 & 0.8 & 0.9 & 1.0
\end{pmatrix}
\]

For both the working correlation matrix structure and the fixed working correlation matrix structure, no estimation of \( \propto \) is performed. For the fixed structure, the user must specify the requirements at \( r \times r \) matrix. The working correlation matrix with the measured data identity and the link function is also referred as compound symmetry, which corresponds to a random effects model with a random intercept per cluster. For the requirement \( (R) \) option, the matrix should be a symmetric matrix with 1’s on the diagonal, which specifies a banded structure with a fixed correlation and linear declines as the distance between observations increases.

4.7 INCORPORATING RISK MANAGEMENT INTO SOFTWARE REQUIREMENT ENGINEERING

Risk is the possibility of loss. It is a function of the probability of an adverse event occurring and its control. Therefore, the impact establishes itself in a combination of economic loss, loss of performance and time delay. Risk can be managed, but cannot be removed from a software project. Risk management is critical to the success of any software effort. The risk management recognizes the possible problem, before they occur. Therefore, risk handling activities can be planned and applied across the life of the
product. It begins at the initial stages of project development and continues throughout the total lifecycle of the project.

Risk management helps to avoid problems, reworks and failure existing in software project for stimulating successful project outcomes. Requirement engineering and software risk management are two different processes.

The incorporation depends on considering the interactions among the techniques, activities, characters, roles and dependencies among the artifacts, which is involved between the two processes. Generally, artifact oriented view focuses on the dependencies among the requirements engineering or risk artifact types. It is a systematic methodology that describes the problem space of the system, it is potential towards complete, reliable and accurate requirements specification documents. It combines both structure and content of the artifacts and is expressed by domain specific concept model. It integrates methods for producing the most reliable and complete result of the artifact.
The artifact oriented requirements engineering focuses on the business information system domain, and it includes business and requirement specification. Figure 4.4 depicts the overview of an artifact types of requirement, business and risk specification. The business specification formulates the proficiencies, aims, conditions and restrictions, which affects the business of customer’s organization.

Moreover, it contains many content items, namely business capabilities, limitations and business vision. The requirement process consists of description, owner, stakeholders, source, time restrictions, attributes id and priority. The risk specification includes content item such as goals, difficulties and management in which concept contains risk details,
risk management plan, and status about risk through textual representation. This process lets supporting the combination of the risk management into the software requirements engineering.

4.8 ESTIMATION OF RISK MANAGEMENT IN SOFTWARE REQUIREMENT ENGINEERING

The risk assessment is an important part of software risk management. In general, there are many techniques for risk assessment during several phases of software development and different levels of abstraction. There are several methods available for assessing risk at the required level. Additionally, it is highly subjective and not accessible for formal design, therefore such methods are highly error prone and human intensive.

Risk is defined as integration of two features.
- Probability of malfunctioning
- Importance of malfunctioning

The proposed work focused on identifying the risk, but it has limited capacity in preventing these risks from occurring. Software requirement engineering risk addresses the possibility of suffering a loss of any functional or non-functional requirement of the software system. Proficient risk analysis would follow a proactive approach, where the possible risks are recognized and protective measures are taken to avoid the risks from occurring. Moreover, identifying requirement risk at this stage is subjective and domain experts investigate the requirements.

4.9 RISK IDENTIFICATION AND EVALUATION IN SOFTWARE REQUIREMENT ENGINEERING

Software risk management is a software engineering practice, which comprises of techniques, processes and tools for managing risks in a project. It affords an environment for proactive decision making to determine the risk and implement action to deal with risk. The risk management plan addresses the approach for risk management process, techniques and tools used to support the risk management development. Figure 4.5 shows the risk management in software requirement engineering.
Figure 4.5 Risk Management in Software Requirement Engineering

- **Identifying the Risk**: Before affecting the project, risk should be identified and managed. While, establishing an environment, the people can increase concerns, issues and managing the quality reviews of a project.

- **Analyzing the Risk**: The software risk evaluation team analyzes each recognized risk in terms of its cost, performance, product quality and schedule. Analysis is the conversion of risk data into risk decision-making information. It contains reviewing, selecting and critical risk to address.

- **Plan for Risks**: It is considered as the important decision to be made. It includes addressing the individual task, creating a risk management plan and prioritizing the risk action.

- **Monitoring the Risk**: This method tracks the status of the risk and actions taken against risks to diminish them.

- **Resolve Risks**: It controls the project management process that controls the risk action plans, accurate plans and finally improves the management process. It is an important part of all other risk management activities.
4.10 RISK ASSESSMENT IN SOFTWARE REQUIREMENT ENGINEERING

Risk assessment is the process of risk management. It is based on three layers of processing risk related information, namely, analysis, identification and reporting. Reviews establish the framework for risk identification, snapshots pass the recognized risks for further analysis and reports communicate the results of risk assessment. This process differs in terms of their duration, scope and identification method. Additionally, risk related information collected during a review is characterized as risk indication, identifies a particular risk, involved the project stakeholders and identifies a possible comments.

The risk assessment report is generated after the identification and analysis of the risk. Risk Summary can be used as input for risk mitigation related activities. Moreover, it can act as an input to the next risk review achievement. The output of the risk assessment process recognizes appropriate controls for reducing risk during the risk mitigation process. The risk identification and analysis process are performed by the project stakeholders and controlled by the risk manager. There are two types of review, namely, active review and continuous review.

- **Active Review:** The starting time and ending time are set by the risk manager along with its scope and participants (the stakeholders involved in the review). The active review ends with the risk analysis session, which aims at calculating and prioritizing the identified risks and produces a significant report.

- **Continuous Review:** The continuous review starts with the end of the previous review and ends with the start of the next review. The set of its input document is not controlled by the risk manager. Several project stakeholders can permit the risk-related information while ignoring the way of its generation.
A snapshot is taken at the end of the continuous review to afford an input to the active review. In addition, snapshots are taken at the end of an active review to summarize the effects of risk identification activities. At the end of an active review, the risk assessment report is generated. Active review and continuous review interleaved, their time, inputs and participants are controlled by the risk manager.

- The identification actions are intended for active review and continuous review
- The communication channel is constantly open
- The recognized risks are periodically assessed and reviewed. Additionally, regularity and scope of these recognized risks are under the control of the risk manager
- All communicated risk related information is being memorized
- The analysis of the results is available downstream of the process.

Risk management in software projects has different uses. This helps to protect projects from failing due to different features, such as budget limitations, non-completion of projects within the identified schedule and not meeting customer expectations. The risk management looks at projects from different views to confirm the threats to the project,
which is recognized, analyzed, controlled and moderated. The mitigation approaches may not be necessary for termination of tasks that include risks. In addition, high-risk tasks are sometimes significant to afford a leading edge over to their opponents in industries.

All possible risks of a project need to measure their severity, and control resolution depending on the risk management. Moreover, appropriate planning leads to minimizing uncertainties or else it might lead to complete cancellation of the projects. Software engineering risk management takes a preventive approach for completion of projects within the budget and predictable time. The risk management project has the ability to increase the quality of the project delivery, timely completion and reduce the project costs. Application of these techniques to the project reduces the loss of profits and customer trust.

4.11 ESTIMATION OF QUALITY REQUIREMENTS

This section focuses on the quality requirement techniques; it is normally defined in terms of universal properties in a software quality system, such as reliability, and scalability. Scalability and reliability are the important attributes, which are mainly used to estimate the stakeholder’s requirements. It is the most significant and most measurable aspect of the software module generation. The quality requirement method is important in inter-system comparability, and these software requirements engineering technique treat as non-functional requirements. The quality requirements are significant throughout the lifecycle and it provides better software requirements after the initial release. In quality requirement, post-release is used to examine the current performs using a standardized requirement taxonomy.

An inefficient software requirements engineering may induce doubt and insecurity in the improvement process, which cause errors in implementing the changes. Consequently, these errors circulate through later phases of development and maintenance. These errors may result in important risks related to implementing the requirements. In reliability risk, the risk of faults and failure changes in requirements acquired by deficiencies in the process.

The software requirements are identified in terms of meeting user expectations and important risk associated with their implementation. While implementing the user requirements, it could lead to system size and difficulty in reliability and scalability. Scalability and reliability measure for various numbers of stakeholders. Consequently,
there is a possibility to obtain reliability and scalability risks, when new software is enhanced in changed requirement. Hence, in measuring the effects of requirements on reliability and scalability, requirements should be dealt throughout the lifecycle. Scalability and reliability is defined as the probability of the failure free software operation for a specified period of time in a particular environment.

4.12 SUMMARY

The requirements engineering process provides the best opportunity to consider all various stakeholder’s interest in context with one another. The research work is focused on coordinating and handling the large number of stakeholders’ requests and arranges the successive requirements into meaningful structures. Here, the Jacquard Similarity measure is used to evaluate the similarity among the stakeholders to cluster the optimal groupings, which is finally used for generating a software model.

Each software development project has unique difficulties, both in terms of customer expectation and project features. This work proposed the software development risk management modeling framework. A module prediction method is proposed to develop a project’s outcome early in the development life cycle. Moreover, it is used to evaluate tradeoffs in software function, quality of results and indicate success of risk factors. The performance of a software can be enhanced by incorporating significant quality attributes like reliability and scalability. Scalability and reliability models describe the failure behavior of the software, which is used to evaluate the software quantitative for a specified period of time in a particular environment. It is the most important and measurable aspect of the software module generation.